AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A network system for key management, comprising:

a server;

a key management system providing process logic for key management system initialization located on the server comprising:

a memory storing data in a n-tuple within the key management system, wherein the n-tuple comprises a key name field, a key value field, and a key type field,

a hashing module hashing a key encryption key,

an encryption module encrypting data, and

a serialization module serializing data obtained from the memory, the encryption module, and the hashing module in order for the data to be stored as an encrypted serial file and persist beyond the time the key management system is active;

a key management system storage providing a secure data storage for the key management system; and

an interface providing a means for inputting data into the key management system, wherein data is used to generate a key in the key management system.

2. (Original) The network system of claim 1, further comprising a client computer operatively connected to the server, wherein the client computer comprises a user interface to input data into the key management system.

3. (Original) The network system of claim 1, wherein the key management storage is located on the server.

4. (Original) The network system of claim 1, wherein the key management storage is located on a second server operatively connected to the server.

5. (Original) The network system of claim 1, wherein the interface comprises a graphical user interface.

6. (Original) The network system of claim 5, wherein the graphical user interface is integrated into a web browser.

7. (Original) The network system of claim 2, wherein the user interface comprises a graphical user interface.

8. (Original) The network system of claim 7, wherein the graphical user interface is integrated into a web browser.

9. (Original) The network system of claim 2, wherein the client computer and the server are connected using an encrypted connection.

10. (Cancelled)

11. (Previously Presented) The network system of claim 1, further comprising:
a randomizer randomizing data.

12. (Previously Presented) The network system of claim 1, further comprising:
an encoding module for encoding data.

13. (Previously Presented) The network system of claim 1, wherein the hashing module uses a MD5 hashing function.

14. (Previously Presented) The network system of claim 1, wherein the encryption module, further comprises a key generation tool.

15. (Previously Presented) The network system of claim 14, wherein the key generation tool comprises a symmetric algorithm.

16. (Previously Presented) The network system of claim 14, wherein the key generation tool comprises an asymmetric algorithm.

17. (Previously Presented) A network system for key management, comprising:
a server;
a key management system providing process logic for key management system initialization located on the server comprising:

a memory storing data in a n-tuple within the key management system, wherein the n-tuple comprises a key name field, a key value field, and a key type field;

a hashing module hashing a key encryption key;

an encryption module encrypting data; and

a serialization module serializing data obtained from the memory, the encryption module, and the hashing module in order for the data to be stored as an encrypted serial file and persist beyond the time the key management system is active;

a key management system storage providing a secure data storage for the key management system;

an interface providing a means for inputting data into the key management system; and

a client computer operatively connected to the server, wherein the client computer comprises a user interface to input data into the key management system,

wherein data is used to generate a key in the key management system.

18. (Currently Amended) A computer implemented method for initializing a key management system comprising:

entering data into a key management system interface, wherein data is used to generate a key in the key management system;

entering a key encryption key into the key management system interface;

combining data into a n-tuple comprising a key name field, a key value field, and a key type field;

encrypting the n-tuple with the key encryption key to produce a secret token;

storing the secret token in a vector;

hashing the key encryption key;

storing a hashed key encryption key in the vector;

storing a list of keys in the vector;

serializing the vector to produce an encrypted serialized file; and

storing the encrypted serialized file in a key management system to persist beyond the time the key management system is active,

wherein the stored encrypted serialized file enables the key and the key encryption key to be securely stored on a server.

19. (Previously Presented) The method of claim 18, further comprising:
encoding a key field of the n-tuple.

20. (Original) The method of claim 19, further comprising:
randomizing the order of the list of encoded keys.

21. (Cancelled)

22. (Original) The method of claim 18, further comprising:
randomizing the order of the list of keys.

23. (Original) The method of claim 18, further comprising:
generating data to encrypt.

24. (Cancelled)

25. (Currently Amended) The method of claim 18, wherein the n-tuple further comprises:
an application name field [[;]]
~~a key field;~~
~~a value field; and~~
~~a type field~~.

26. (Original) The method of claim 18, wherein the vector comprises:
a secret token portion;
a key encryption key hash portion; and
a key list portion.

27. (Currently Amended) The method of claim 26, further comprising [[;]] :
tagging the secret token ~~with an application name~~ to associate it with an application, wherein the tag comprises the application's name.

28. (Original) The method of claim 26, further comprising:
tagging the key in the key list with an application name.

29. (Original) The method of claim 18, wherein the key management storage is located on a second server.

30. (Original) The method of claim 18, wherein the key management system interface comprises a graphical user interface.

31. (Original) The method of claim 30, wherein the graphical user interface is integrated into a web browser.

32. (Original) The method of claim 18, wherein the encrypting comprises using a symmetric algorithm.

33. (Original) The method of claim 18, wherein the encrypting comprises using an asymmetric algorithm.

34. (Currently Amended) A computer implemented method for initializing a key management system comprising:

entering data into a key management system interface, wherein data is used to generate a key in the key management system;

entering a key encryption key into the key management system interface;

combining data into a n-tuple comprising a key name field, a key value field, and a key type field;

encrypting the n-tuple with the key encryption key to produce a secret token;

storing the secret token in a vector;

hashing the key encryption key;

storing a hashed key encryption key in the vector;

storing a list of keys in the vector;

serializing the vector to produce an encrypted serialized file;

storing the encrypted serialized file in a key management system storage to persist beyond the time the key management system is active;

encoding a key field of the n-tuple;

randomizing the order of the list of keys; and

generating data to encrypt.

wherein the stored encrypted serialized file enables the key and the key encryption key to be securely stored on a server.

35. (Currently Amended) An apparatus for initializing a key management system comprising:

means for entering data into a key management system interface, wherein data is used to generate a key in the key management system;

means for entering a key encryption key into the key management system interface;

means for combining data into a n-tuple comprising a key name field, a key value field, and a key type field;

means for encrypting the n-tuple with the key encryption key to produce a secret token;

means for storing the secret token in a vector;

means for hashing the key encryption key;

means for storing a hashed key encryption key in the vector;

means for storing a list of keys in the vector;

means for serializing the vector to produce an encrypted serialized file;

means for storing the encrypted serialized file in a key management system storage to persist beyond the time the key management system is active, wherein the stored encrypted serialized file enables the key and the key encryption key to be securely stored on a server;

means for encoding a key field of the n-tuple;

means for randomizing the order of the list of keys; and

means for generating data to encrypt.